



Decision Making in (multi) Robot Systems

Olivier Simonin

► To cite this version:

Olivier Simonin. Decision Making in (multi) Robot Systems. Doctoral. GdR Robotics Winter School: Robotica Principia, Centre de recherche Inria Sophia Antipolis – Méditerranée, France. 2019, pp.1-50. cel-02130231

HAL Id: cel-02130231

<https://inria.hal.science/cel-02130231>

Submitted on 15 May 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial| 4.0 International License

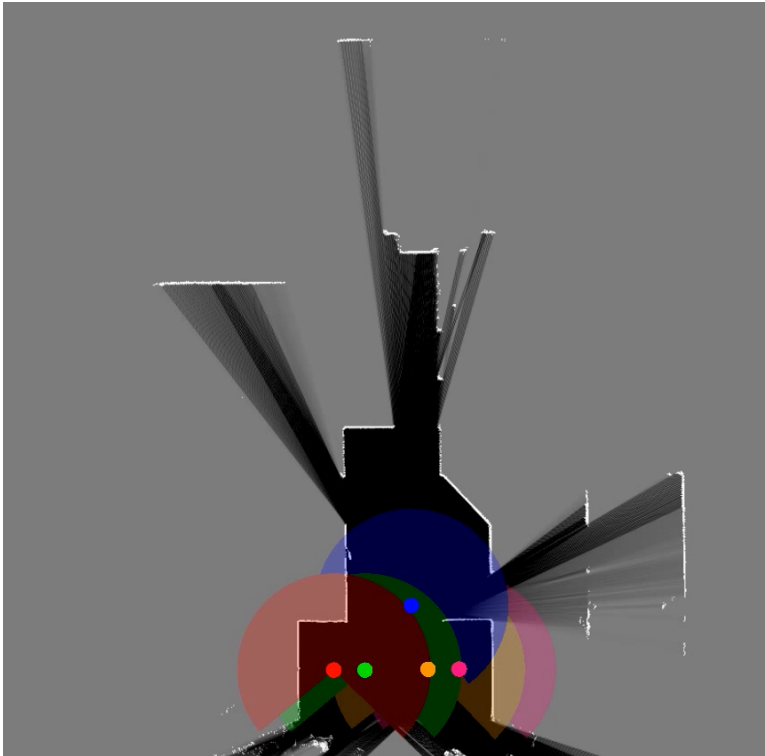
Decision Making in (multi) Robot Systems

Olivier Simonin

INSA Lyon – CITI Lab. - Inria Chroma team



Illustration ..



ANR Carotte Challenge - Final 2012
Cartomatic team

Outline

- I. Decision making ?
- II. Classical architectures
- III. Learning based architectures
- _____
- IV. Decision in multi-robot systems
- V. Strategies for multi-robot exploration

Outline

- I. Decision making ?
- II. Classical architectures
- III. Learning based architectures
- _____
- IV. Decision in multi-robot systems
- V. Strategies for multi-robot exploration

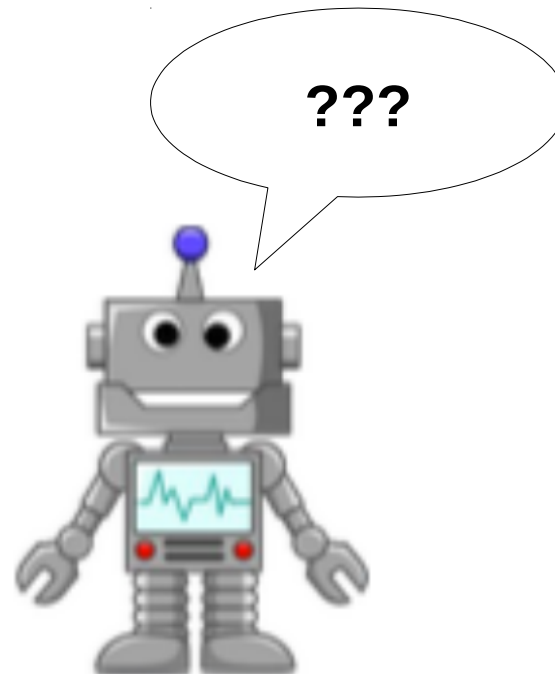
Decide what ?

Actions to fulfill my task

To avoid collisions / risks / breakdown

To cooperate with other robots

...



Decide what ?

Actions to fulfill my task

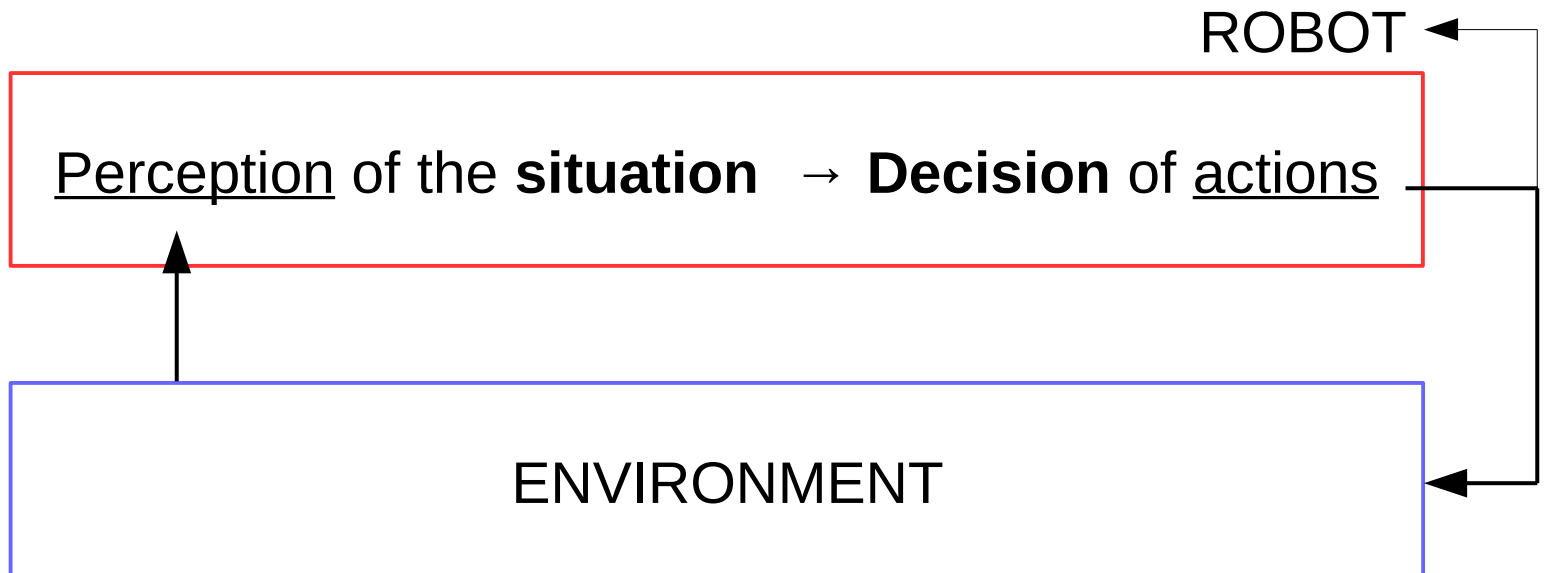
To avoid collisions / risks / breakdown

To cooperate with other robots

...

Perception of the **situation** → **Decision** of actions

Loop Perception-Decision-Action



Loop Perception-Decision-Action

ROBOT

Perception of the **situation** → **Decision** of actions

A.I.

Control

Loop Perception-Decision-Action

ROBOT

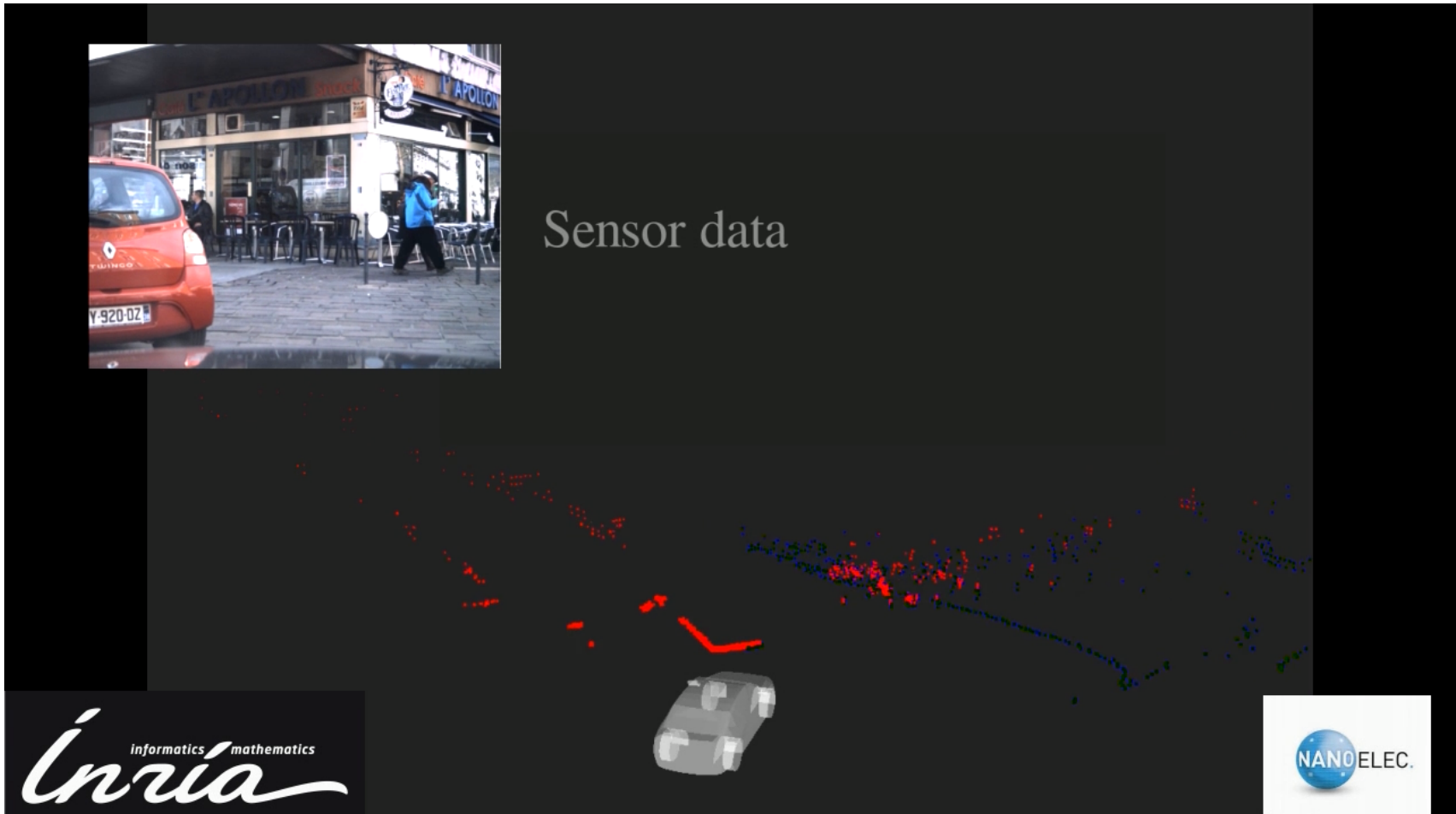
Perception of the **situation** → **Decision** of actions

Modeling the environment

Situation awareness

Predict

Ex.1 Situation awareness with probabilistic grid



Occupancy grid + veloc. + Bayesian F.

[Laugier et al. 2012-2018]

Ex.1 Situation awareness with probabilistic grid

Bayesian Occupancy Filtering (BOF¹, CMCDOT²)

Occupancy grid

Velocity distribution / cell

Object identification (cell clustering)

Prediction of motion (bayes. filtering)

Occupancy grid :

Probability of occupancy in each cell, $P_{occ}(x,y)$

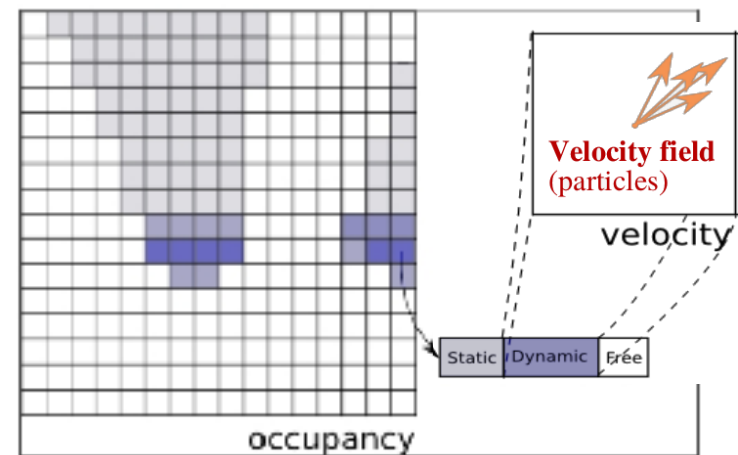
1 : occupied cell (black)

0.5 : unknown (gray)

0 : free cell (white)

e.g. Frequency approach (counting) :

$$P_{occ}(x,y) = occ(x,y) / (empty(x,y) + occ(x,y))$$



¹ C. Laugier et al., IJRR 2005, ² Rummelhardt et al. ITS 2015

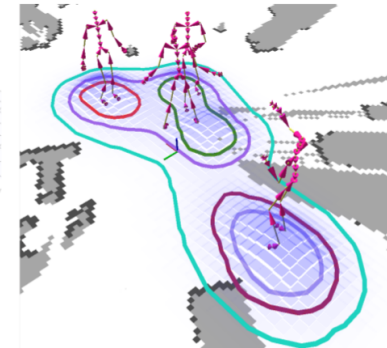
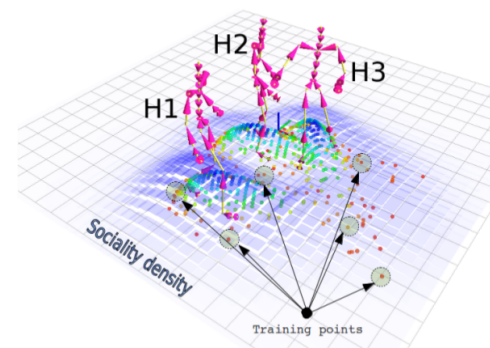
Ex.2 Social navigation : Proxemics approach

Identify humans and **predict** their motion

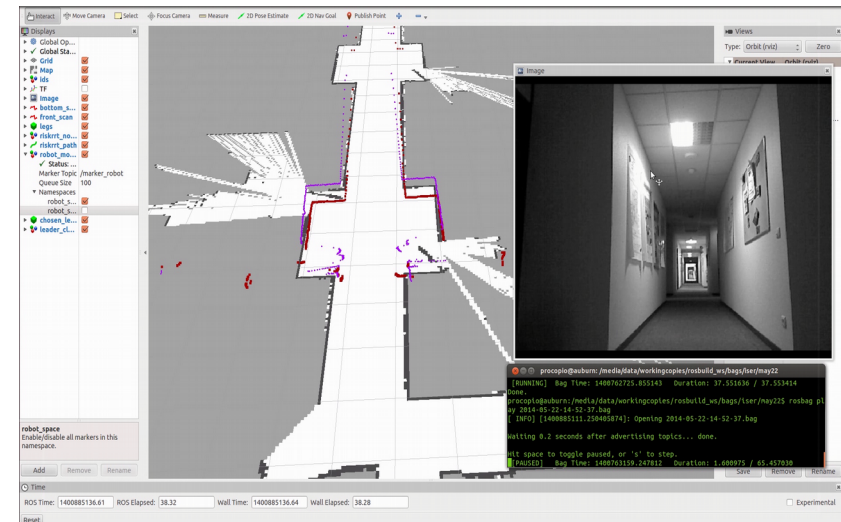
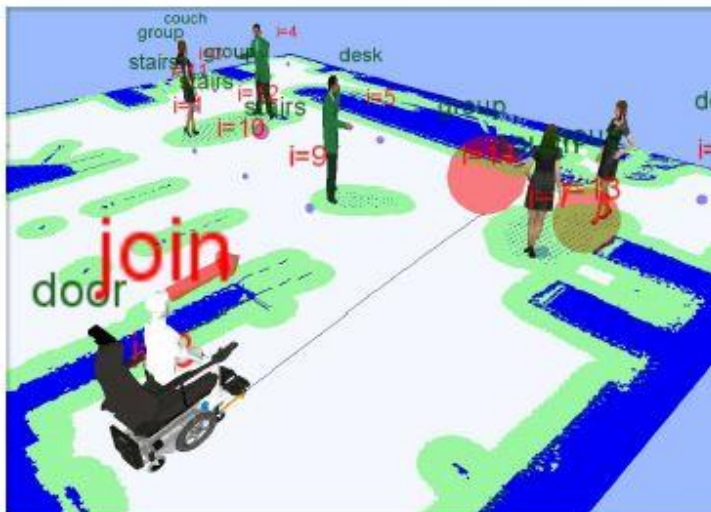
Comply with **social rules**

Map human activities

Compute **secure** paths and decisions



Gaussian model of personal area + geometric formation



Loop Perception-Decision-Action

ROBOT

Perception of the **situation** → **Decision** of actions

Reacting

Planning / Learning

Reacting vs. Planning vs. Learning

Reacting : compute **one action** (real time)

Planning : compute a **sequence** of actions (eg. motion planning)

Learning : compute a **policy** (state \rightarrow action)

+

Cooperate/compete : add **other agents** in the decision

Reacting vs. Planning vs. Learning

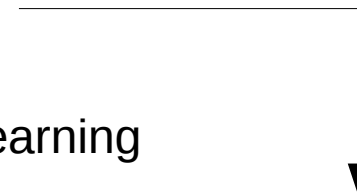
Reacting : Bio-inspired behaviors, Connexionism

Planning : STRIPS, PDDL, SAT, CSP, PRM ..

Learning : Reinforcement L., (PO)MDP, RNN, DeepLearning

+

Cooperate/compete : add other agents in the decision

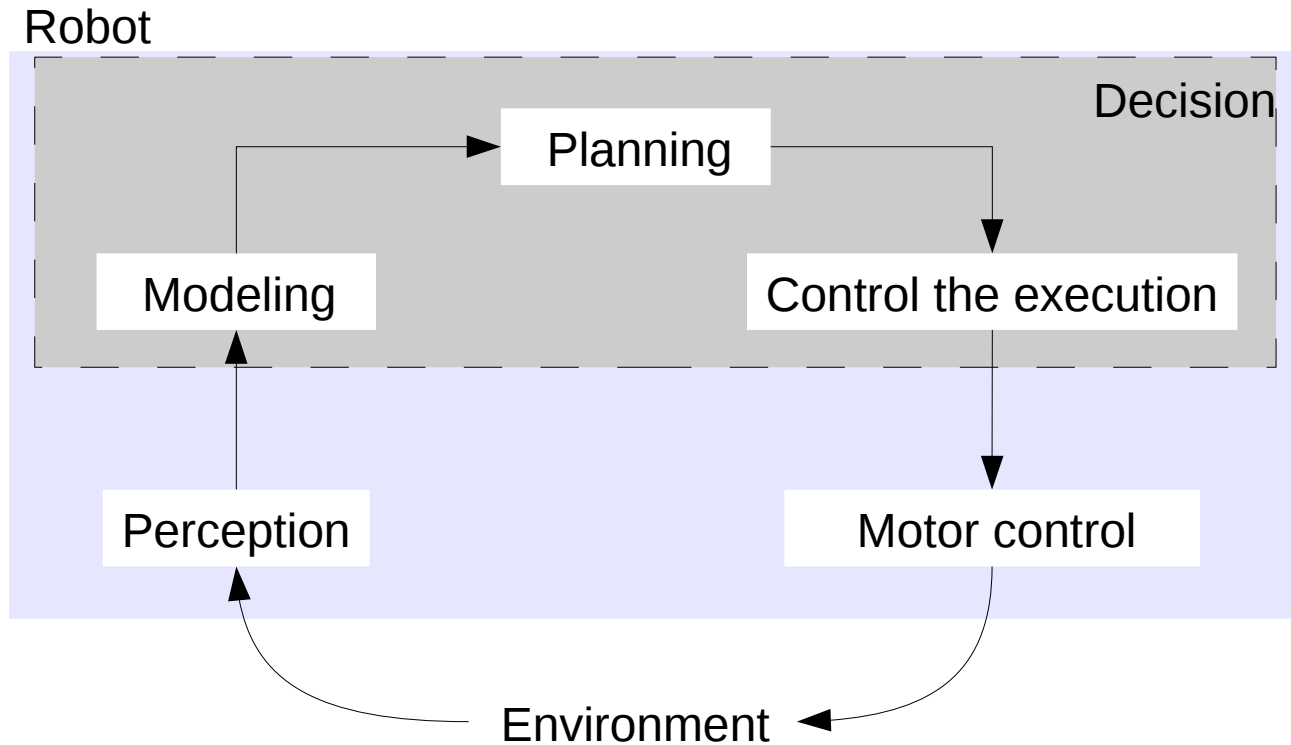


Outline

- I. Decision making ?
- II. Classical architectures
- III. Learning based architectures
- _____
- IV. Decision in multi-robot systems
- V. Strategies for multi-robot exploration

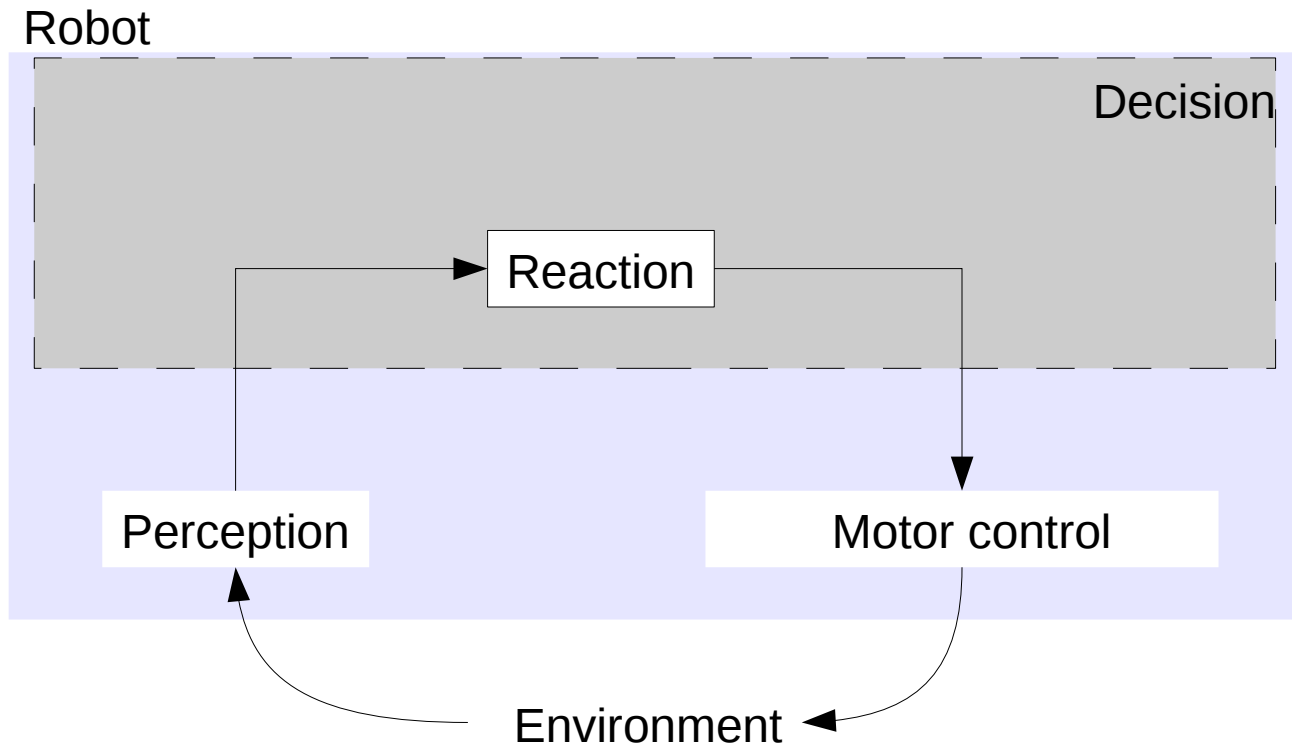
Sense-Plan-Act

[Nilsson80]



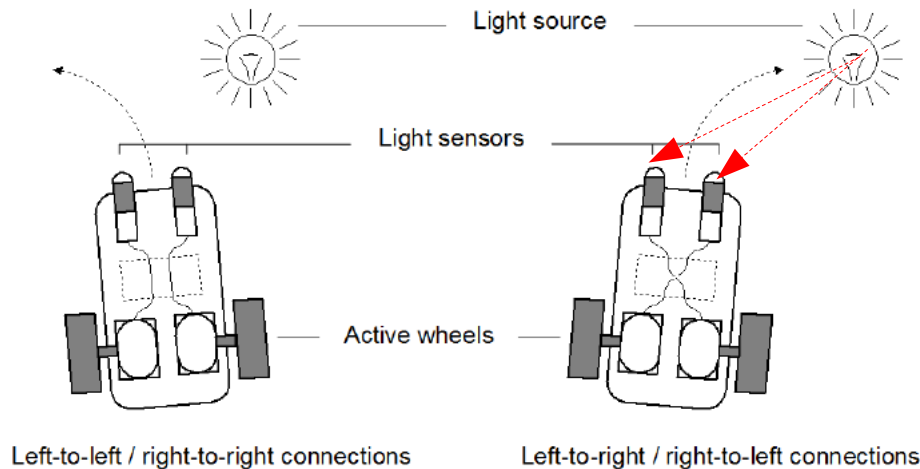
Slow planning → bottleneck !

Reactive architecture

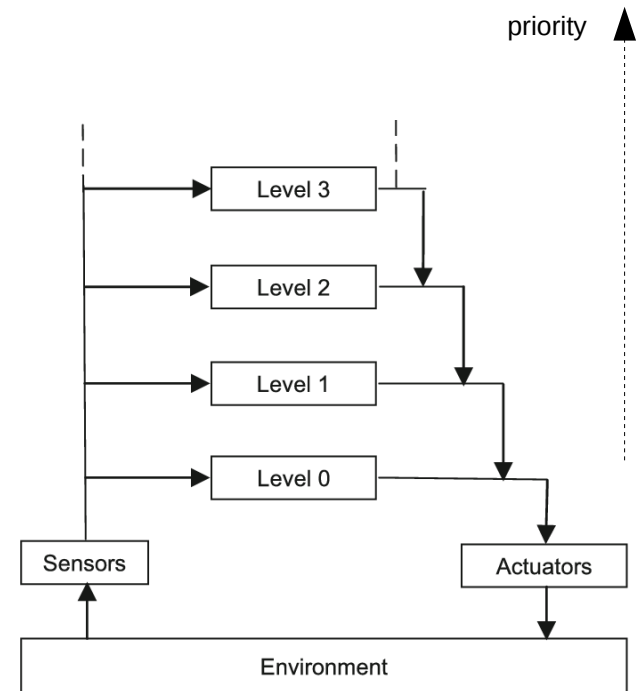


[Brooks 86] [Arkin 98]. Connecting sensors-motors, but **deadlocks** are possible !

Reactive architecture : Braitenberg, Brooks



[Braitenberg 84] Vehicle (phototaxis)

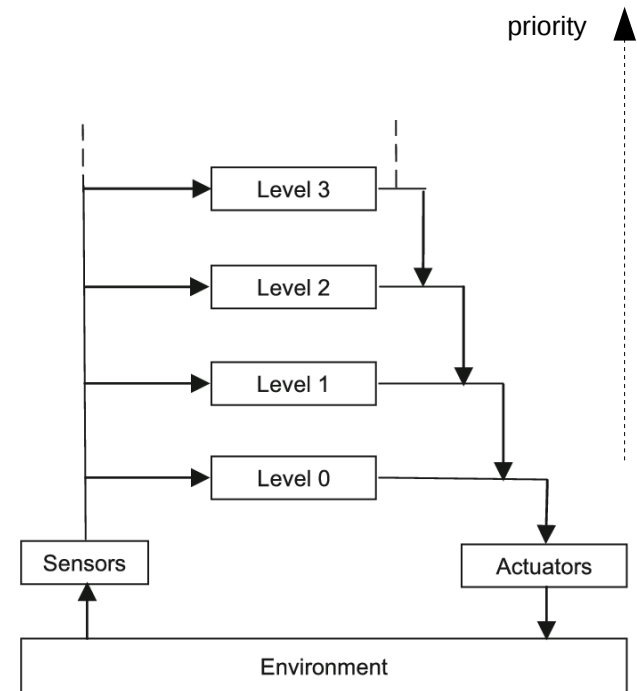


[Brooks 86] Subsumption architecture

Reactive architecture : Braitenberg, Brooks

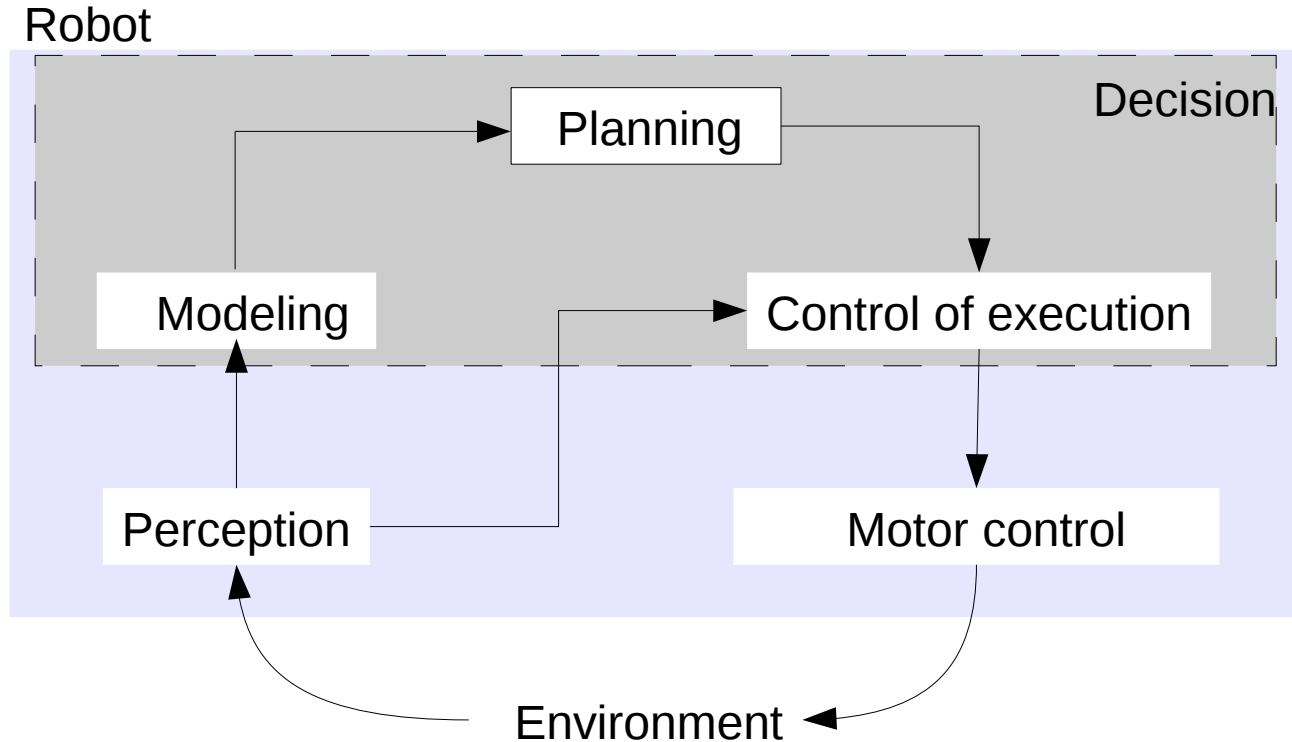


[Braitenberg 84] Vehicle (phototaxis)



[Brooks 86] Subsumption architecture

Multi-level architecture

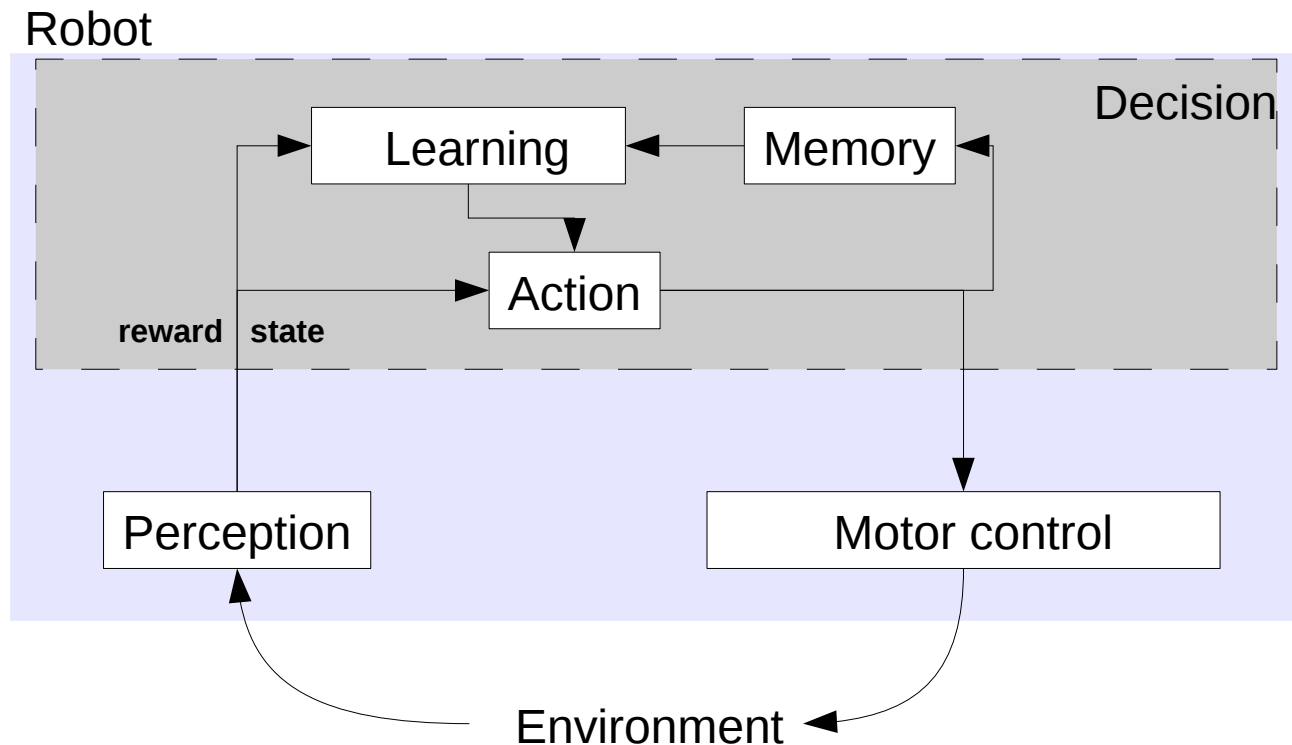


Allow to combine fast reaction and planning

Outline

- I. Decision making ?
- II. Classical architectures
- III. Learning based architectures
- IV. Decision in multi-robot systems
- V. Strategies for multi-robot exploration

Learning architecture : non explicit knowledge repr.



Neuronal networks, Reinforcement Learning, Markov Decision Process (MDP) ..

Q-Learning [Watkins 89] (Reinforcement Learning)

Compute the **quality of an action a in state s** ($s \in S, a \in A$)

$$Q : S \times A \rightarrow \mathbb{R}$$

Each time step t the agent selects an action a_t , **observes a reward r_t** , enters in new state s_{t+1} .

Then $Q(s,a)$ is updated :

$$Q^{new}(s_t, a_t) \leftarrow (1 - \alpha) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)}^{\text{learned value}}$$

Converge to optimal policy that **maximizes the expected cumulative reward** $\sum_{t=0}^{\infty} \gamma^t r_t$
(proof [Watkins, Dayan 92])

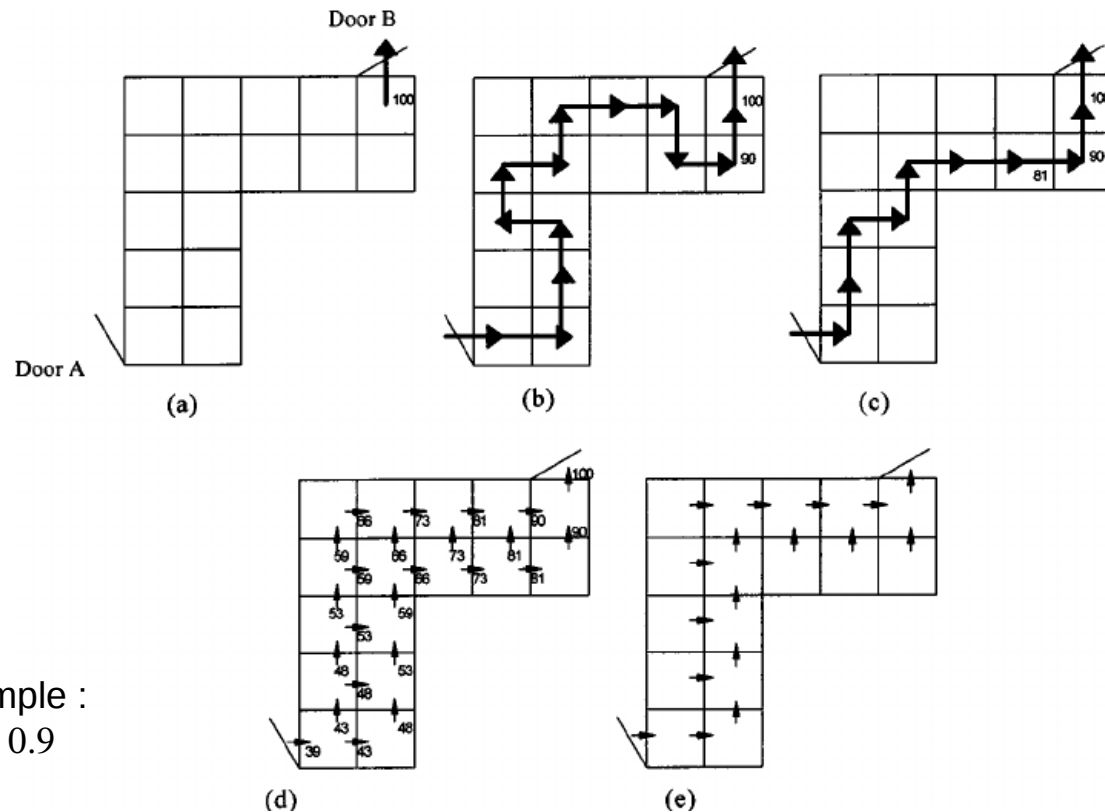
Q-Learning [Watkins 89] (Reinforcement Learning)

$$Q^{new}(s_t, a_t) \leftarrow (1 - \underbrace{\alpha}_{\text{learning rate}}) \cdot \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \left(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} \right)$$

learned value

The agent must explore/experiment the environment

→ exploration/exploitation dilemma



Grid example :
 $\alpha = 1$ $\gamma = 0.9$

Modeling uncertainty : Markov Decision Process

MDP : 4-tuple (S, A, P_a, R_a)

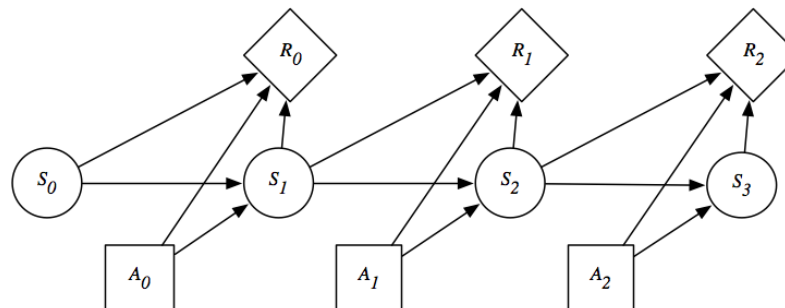
Actions' result is uncertain : that is true in robotics !

Exploit a **model of the environment/system dynamics** :

The probability that action \mathbf{a} , in state \mathbf{s} at time \mathbf{t} will lead to state \mathbf{s}' at time $\mathbf{t+1}$ is

$$P_a(s,s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$$

Each action \mathbf{a} , from state \mathbf{s} to state \mathbf{s}' gives an immediate **reward** $R_a(s,s') \in \mathfrak{R}$



Modeling uncertainty : Markov Decision Process

MDP : 4-tuple (S, A, P_a, R_a)

Actions' result is uncertain : that is true in robotics !

Exploit a [model of the environment/system dynamics](#) :

The probability that action \mathbf{a} , in state \mathbf{s} at time \mathbf{t} will lead to state \mathbf{s}' at time $\mathbf{t+1}$ is

$$P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$$

Each action \mathbf{a} , from state \mathbf{s} to state \mathbf{s}' gives an immediate [reward](#) $R_a(s, s') \in \mathbb{R}$

Problem : finding a **policy** π that maximizes a cumulative function of the random rewards

$$\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1})$$

$$a_t = \pi(s_t)$$

→ **Reinforcement Learning**

Modeling uncertainty : Markov Decision Process

MDP : 4-tuple (S, A, P_a, R_a)

Actions' result is uncertain : that is true in robotics !

Exploit a [model of the environment/system dynamics](#) :

The probability that action \mathbf{a} , in state \mathbf{s} at time \mathbf{t} will lead to state \mathbf{s}' at time $\mathbf{t+1}$ is

$$P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$$

Each action \mathbf{a} , from state \mathbf{s} to state \mathbf{s}' gives an immediate [reward](#) $R_a(s, s') \in \mathfrak{R}$

Problem : finding a **policy** π that maximizes a cumulative function of the random rewards

Several solutions :

- Value iteration [Bellman 57]
- Policy iteration [Howard 60]
- ..

$$\pi(s) := \arg \max_a \left\{ \sum_{s'} P(s' | s, a) (R(s' | s, a) + \gamma V(s')) \right\}$$

$$V(s) := \sum_{s'} P_{\pi(s)}(s, s') (R_{\pi(s)}(s, s') + \gamma V(s'))$$

Computation can be very expensive..

Modeling uncertainty : Markov Decision Process

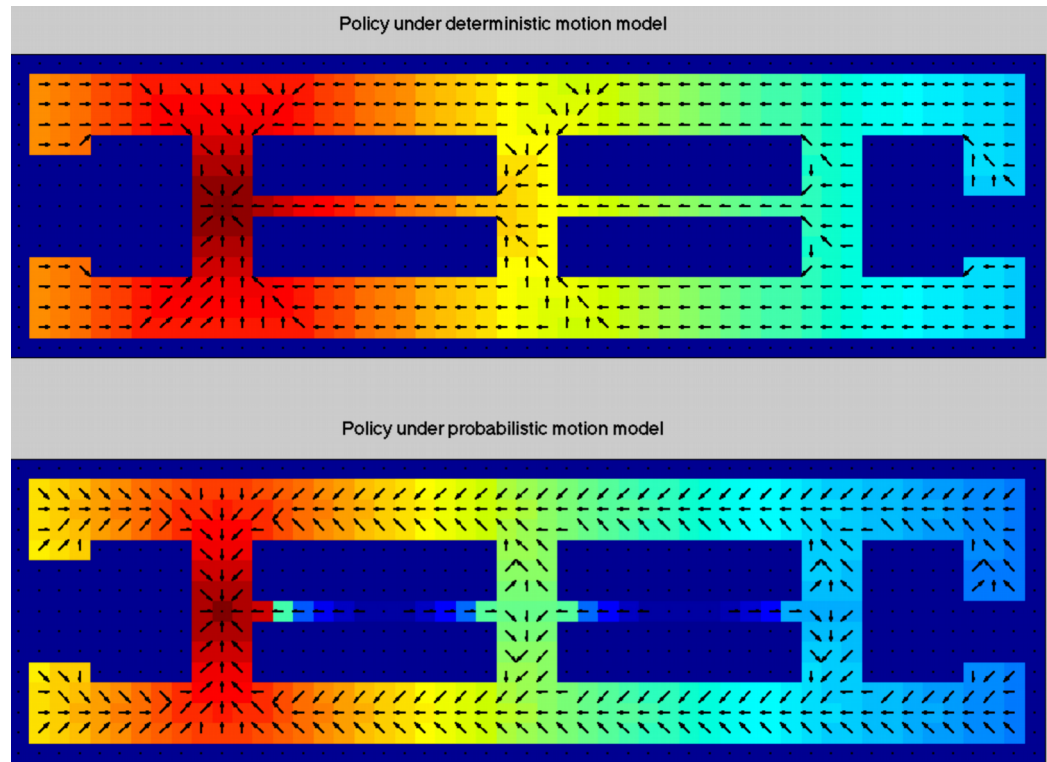
MDP : 4-tuple (S, A, P_a, R_a)

Actions' result is uncertain : that is true in robotics !

Exploit a [model of the environment/system dynamics](#) :

The probability that action a , in state s at time t will lead to state s' at time $t+1$ is

$$P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$$



Modeling uncertainty : Markov Decision Process

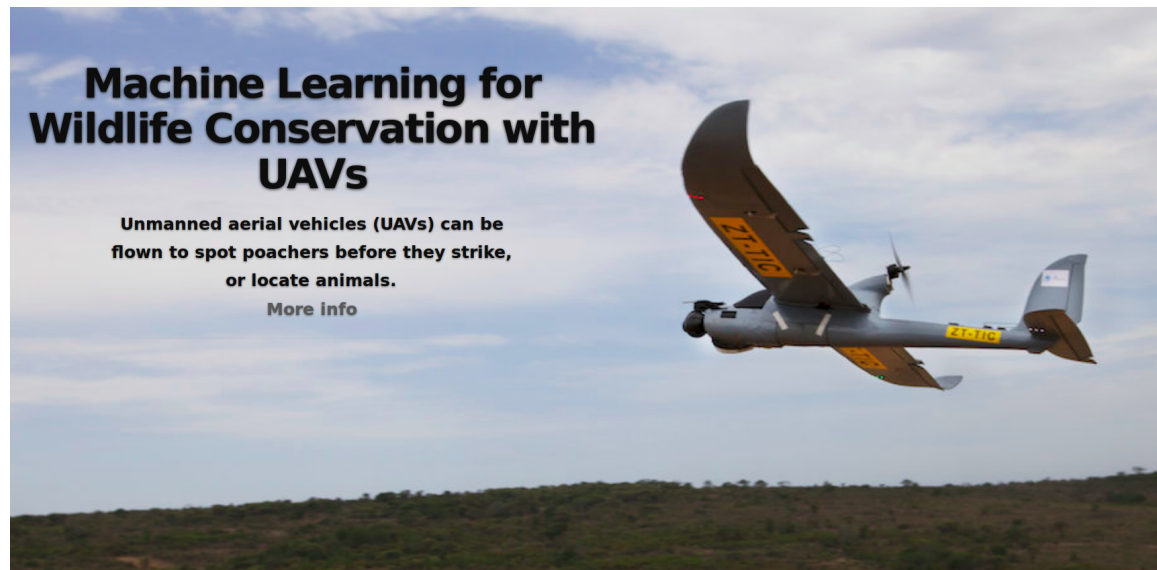
MDP : 4-tuple (S, A, P_a, R_a)

Actions' result is uncertain : that is true in robotics !

Exploit a [model of the environment/system dynamics](#) :

The probability that action a , in state s at time t will lead to state s' at time $t+1$ is

$$P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$$



Good introduction :

Reinforcement Learning: a Survey

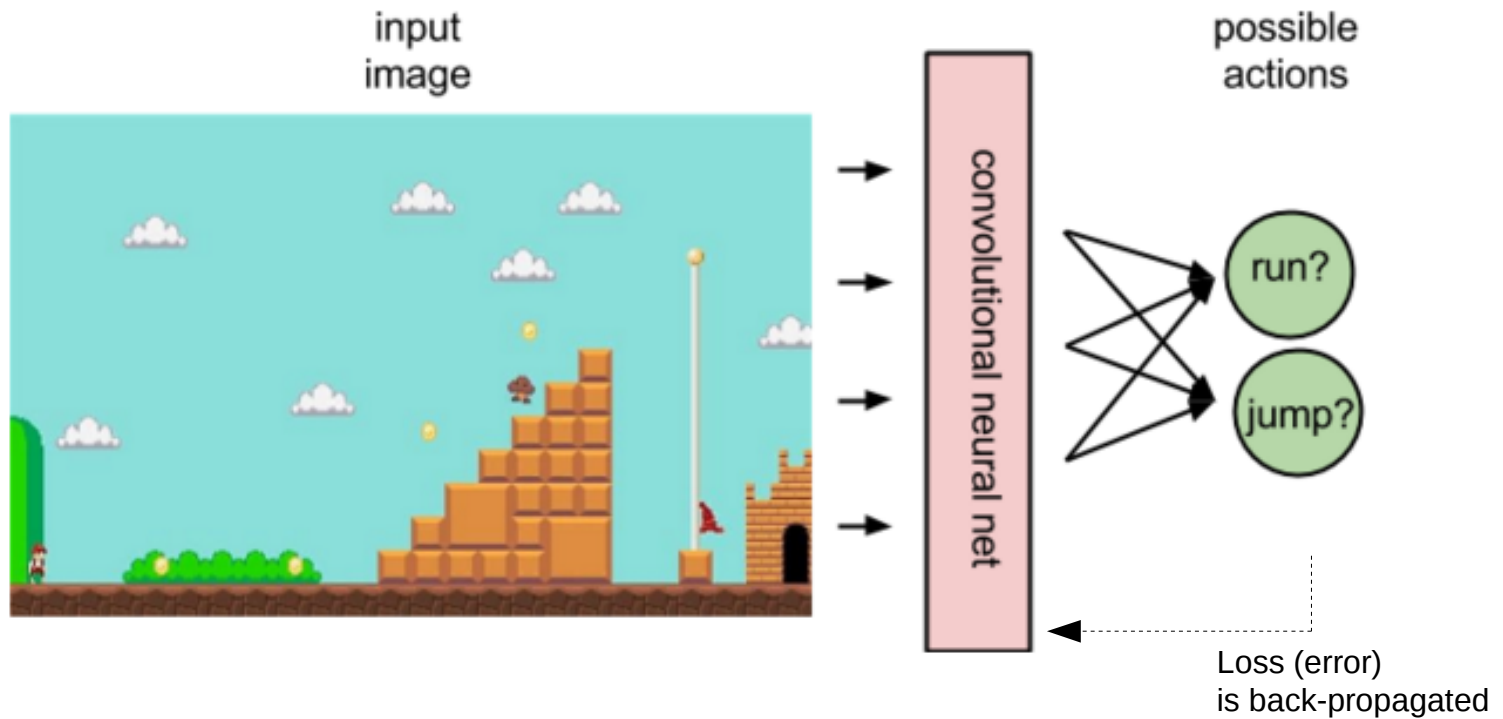
L. P. Kaelbling, M. L. Littman, W. Moore
1996.

Milind Tambe team USC, CAIS

End to End Deep Learning

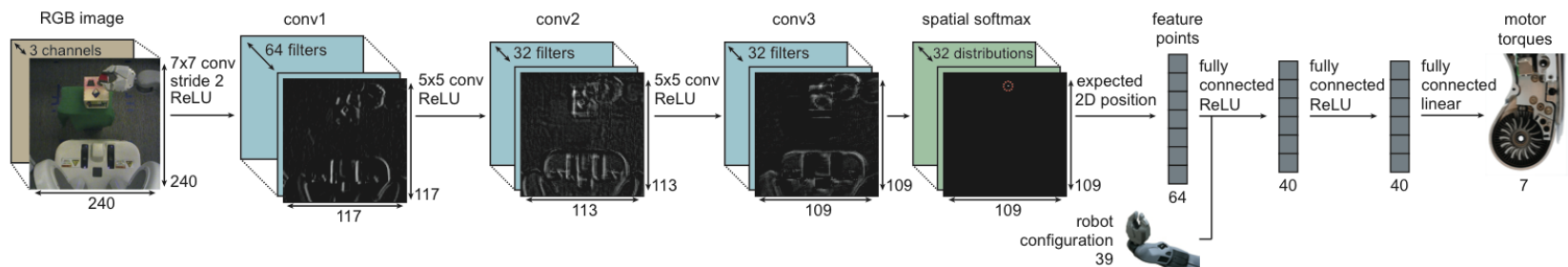
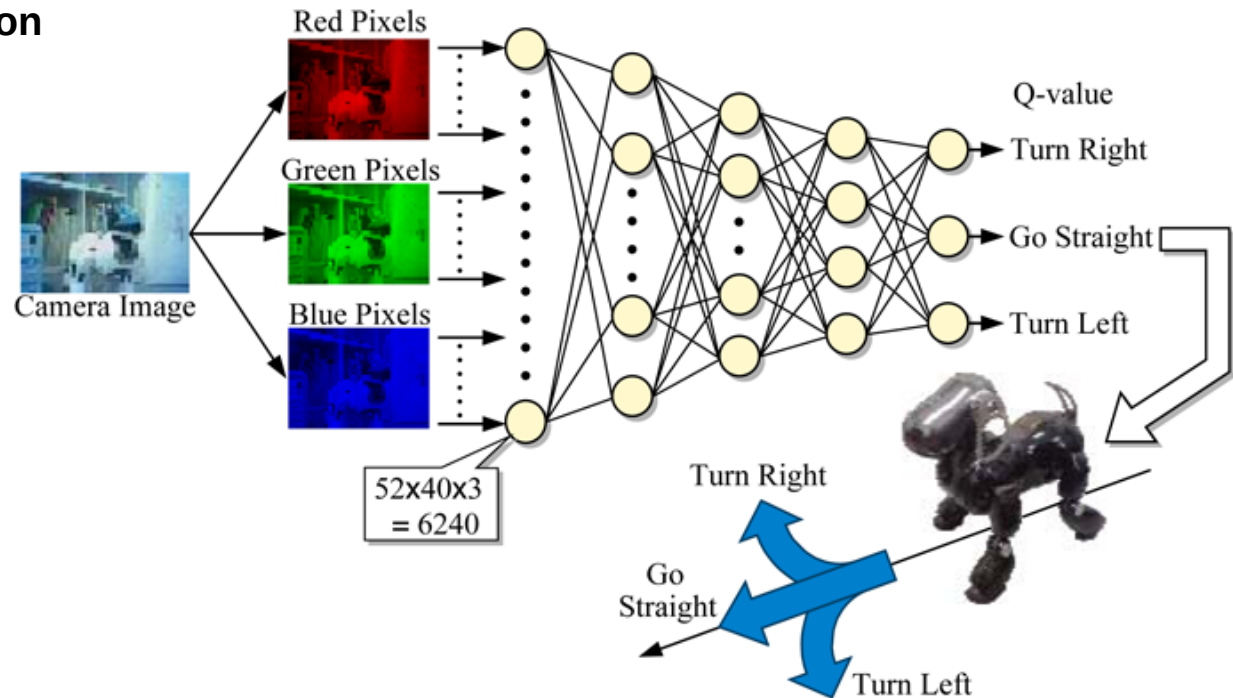
n processes → 1 single Neural Network

Convolutional Agent



End to End Deep Learning

CNN allows generalization

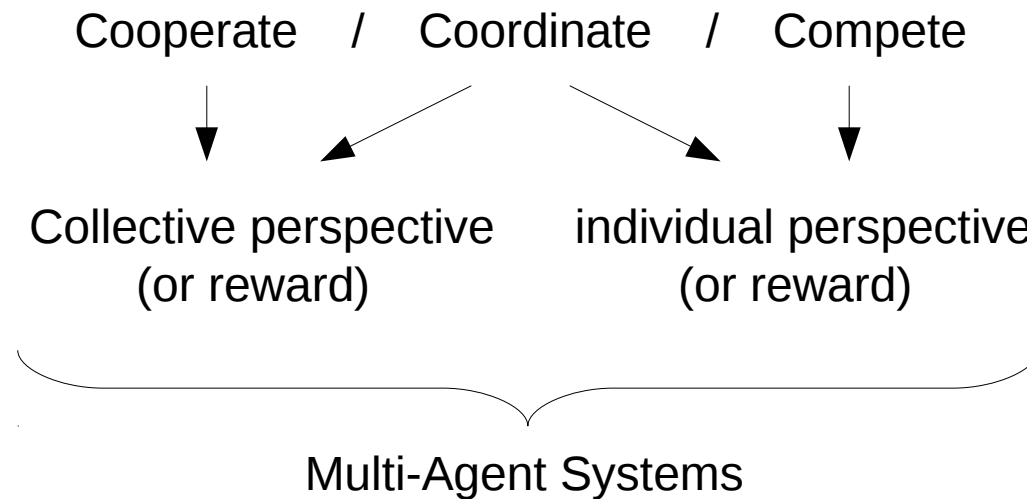


Outline

- I. Decision making ?
- II. Classical architectures
- III. Learning based architectures
- IV. Decision in multi-robot systems
- V. Strategies for multi-robot exploration



Multi-Agent Systems (MAS)



*Autonomous agents that interact in a common environment
in order to fulfill collective and/or individual objectives*

J. Ferber, Multi-Agent Systems, Addison Wesley, London, 1999

Multi-robot missions

EXPLORATION

Mapping (eg. SLAM)

Search, Coverage, Patrolling

Tracking, Active perc.

OBSERVATION

TRANSPORT

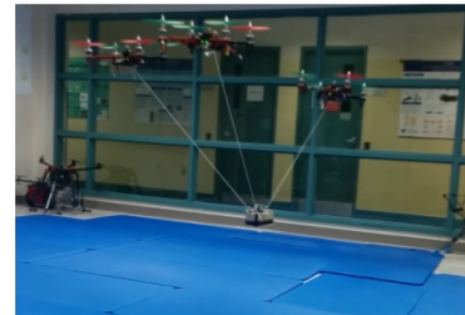
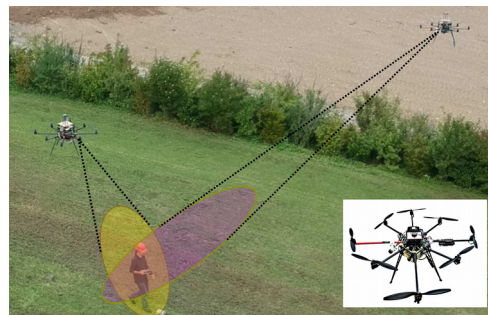
Collective Trans.
(eg. box pushing)

Traffic regulation

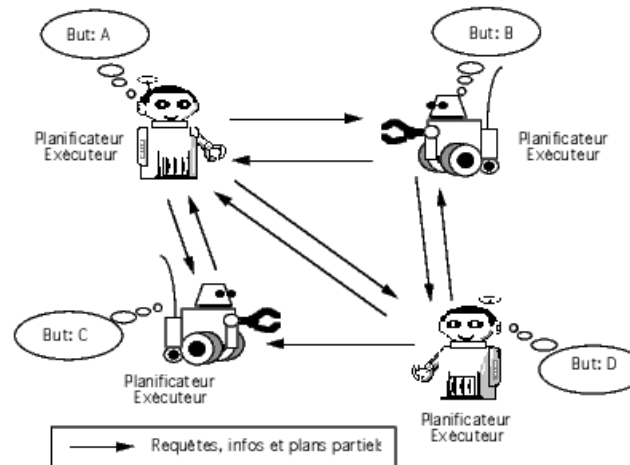
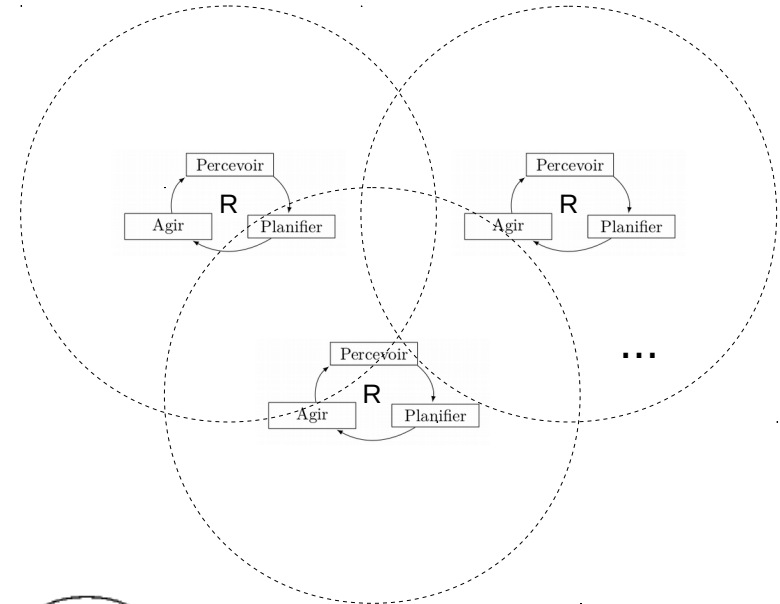
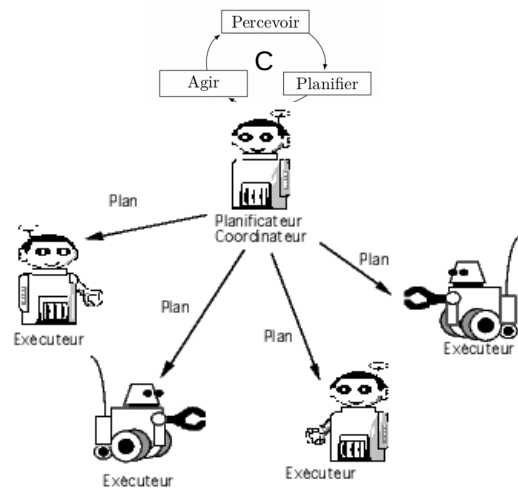
Autonomous fleet navig.

Connectivity

FORMATION MAINT.



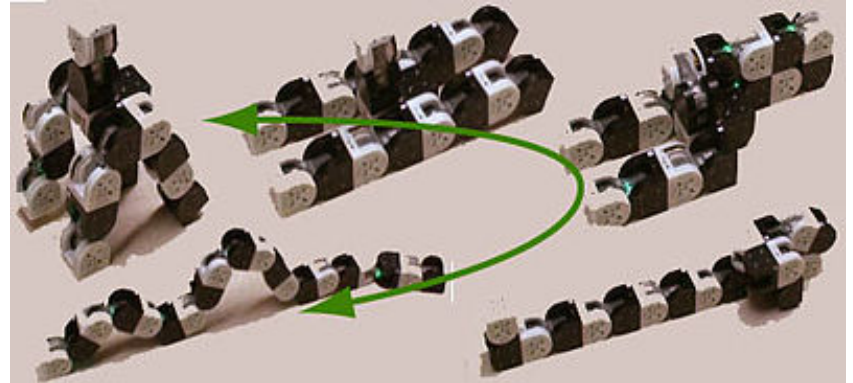
Centralized, Decentralized and Distributed systems



Decentralized : collective navigation and self-config.



Box-pushing 1988



Self-configurable robots (M-TRAN III) 2005



Kilobots (2011)

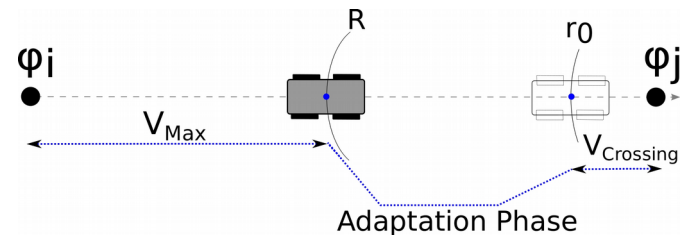
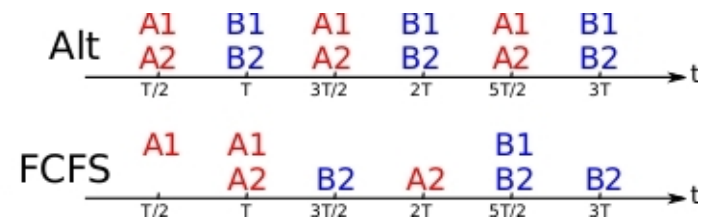
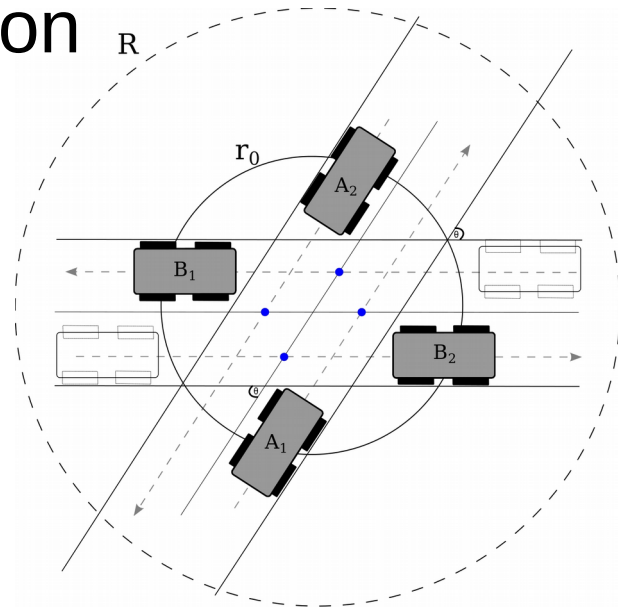


CollMot project (UAV **flocking**) Pennsylvania 2012

Local coordination at intersection

Non stop strategies

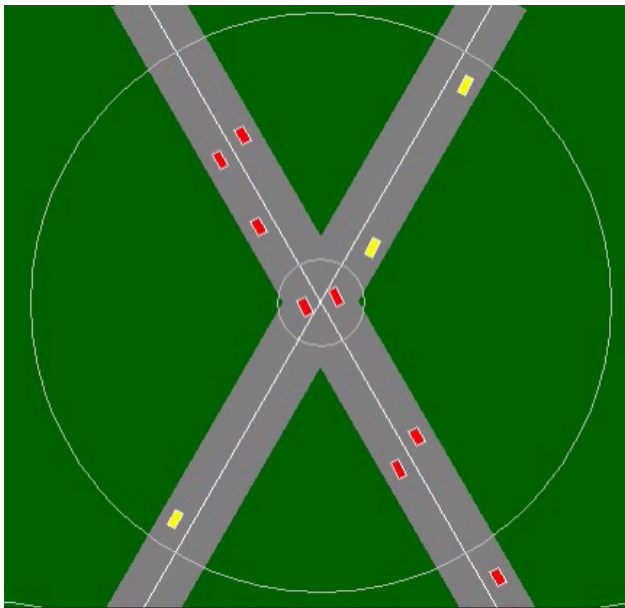
- First Come First Serve (**FCFS**)
[Dresner Stone, 2005]
- **Alternate** [Tlig PhD, 2015]



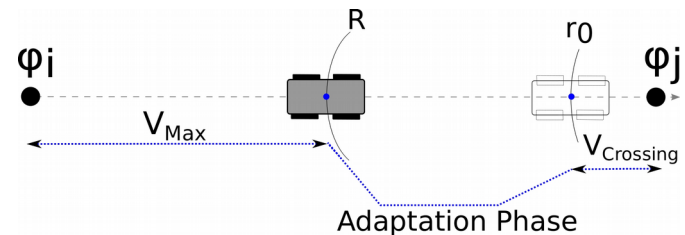
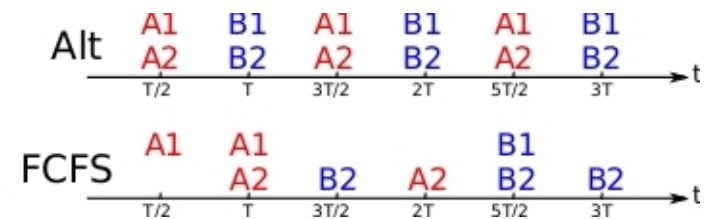
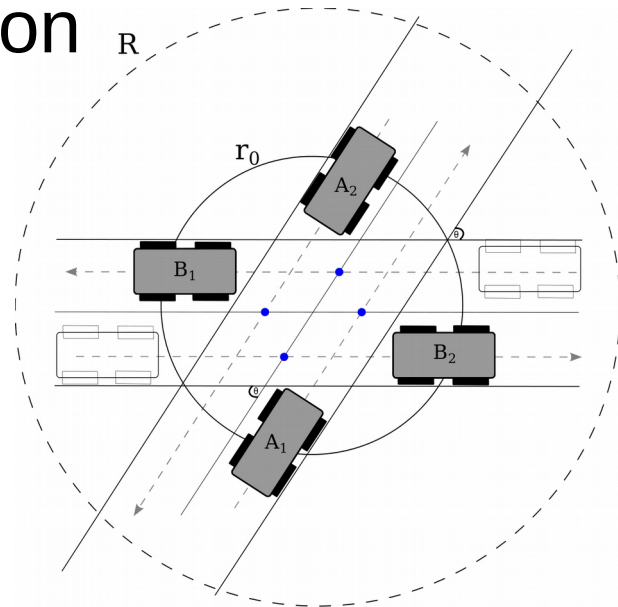
Local coordination at intersection

Non stop strategies

- First Come First Serve (**FCFS**)
[Dresner Stone, 2005]
- **Alternate** [Tlig PhD, 2015]



Tlig, Buffet, Simonin, ECAI 2014



Local coordination at intersection

Cooperation between intersections

Decentralized (global) optimization

→

emergence of **green waves**

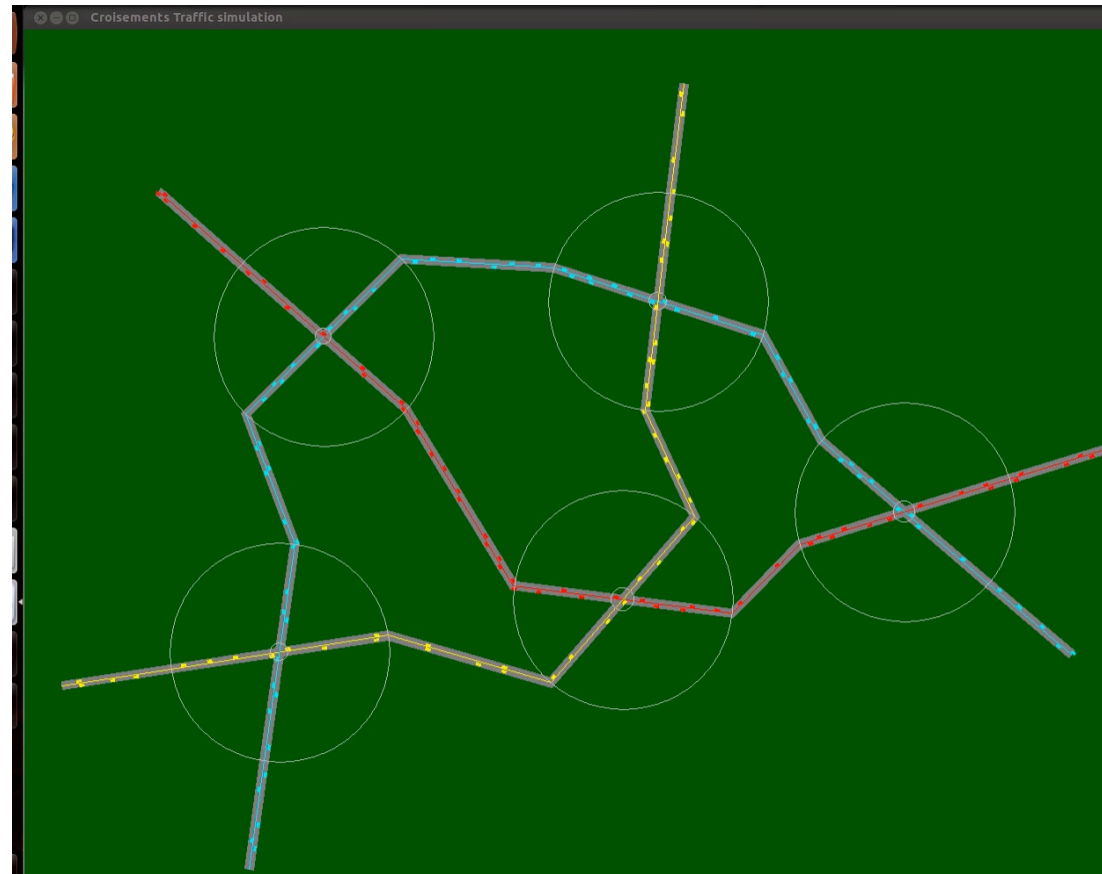
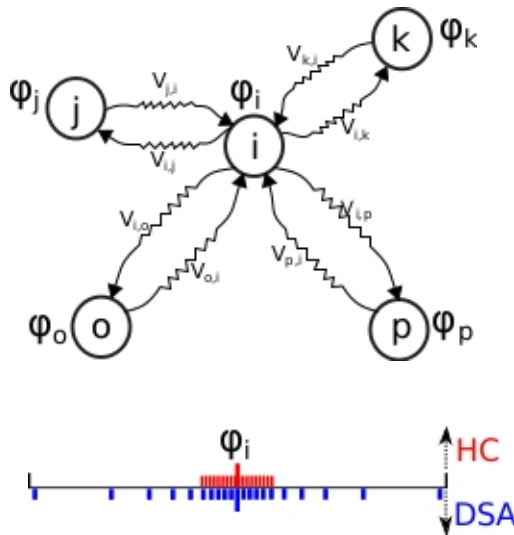
parameters : temporal phases ϕ_i

Optimizing time / energy

→

online adaptation to traffic changes

Hill Climbing, ρ -DSA

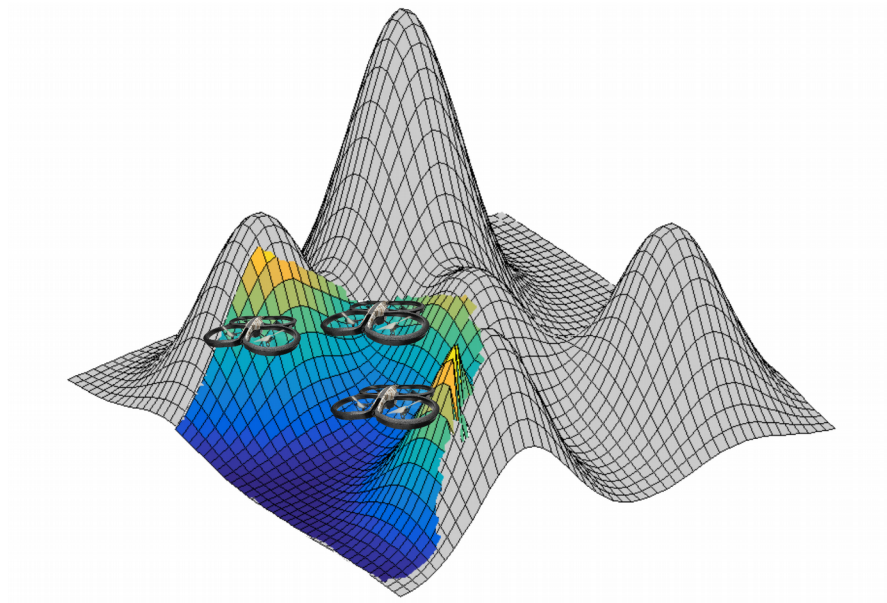


Outline

- I. Decision making ?
 - II. Classical architectures
 - III. Learning based architectures
-
- IV. Decision in multi-robot systems
 - V. Strategies for multi-robot exploration

3D Mapping with a fleet of drones

A. Renzaglia, J. Dibangoye, O. Simonin

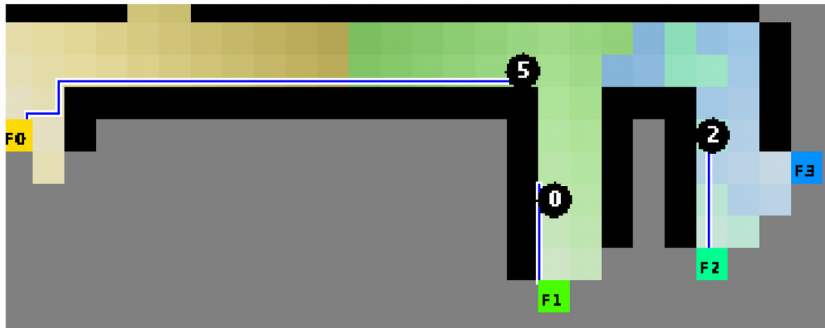


Strategy of exploration: approaches and limits

Mapping relies on visiting frontiers (known-unknown areas)

→ Repeat assignation robot-frontier : $\min \sum_{(i,j)} \text{cost-Dist}(r_i, f_j)$

[Yamauchi98] (mindist), [Burgard05] (greedy), [Bautin12] (minpos) ..

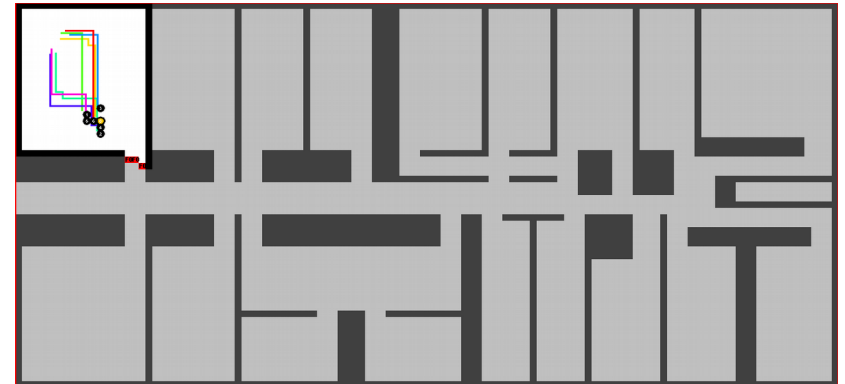
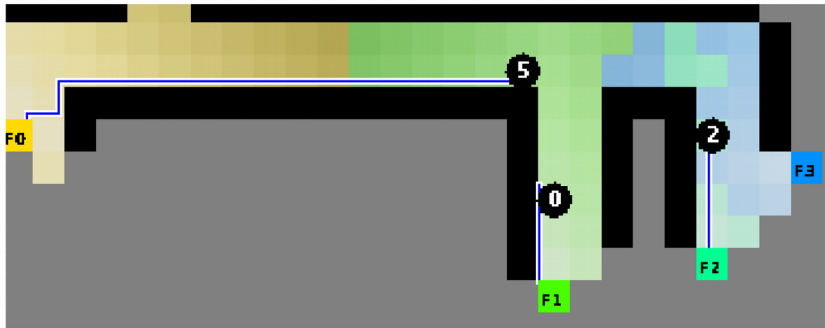


Strategy of exploration: approaches and limits

Mapping relies on visiting frontiers (known-unknown areas)

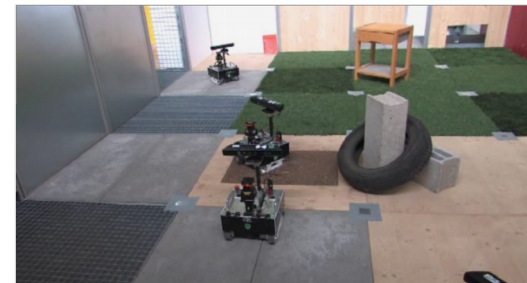
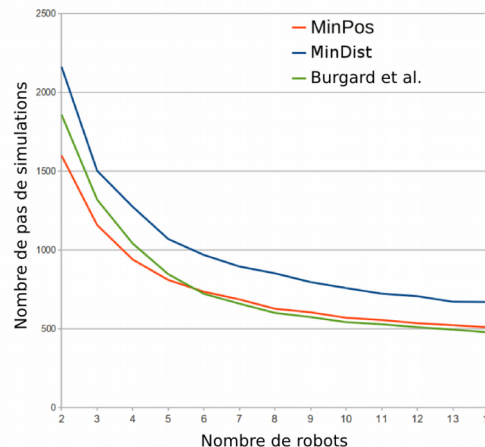
→ Repeat assignation robot-frontier : $\min \sum_{(i,j)} \text{cost-Dist}(r_i, f_j)$

[Yamauchi98] (mindist), [Burgard05] (greedy), [Bautin12] (minpos) ..



Multi-robot exploration : MinPos

[Bautin, Simonin, Charpillet, 2012], ANR Cartomatic



3D mapping : exploit optimal coverage

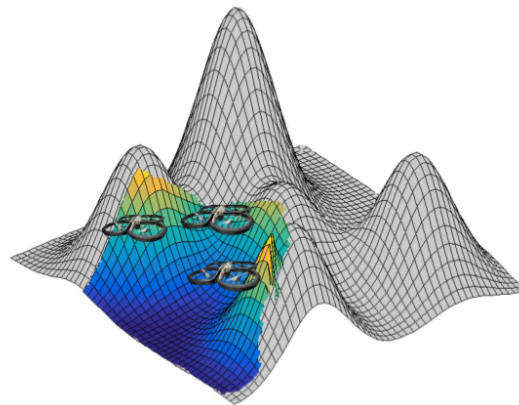
Mapping relies on visiting frontiers (known-unknown areas)

→ Repeat assignation robot-frontier : $\min \sum_{(i,j)} \text{cost-Dist}(r_i, f_j)$

[Yamauchi98] (mindist), [Burgard05] (greedy), [Bautin12] (minpos) ..

Sub-problem: **optimal coverage**

→ deploying robots such as **maximizing the observed area**



3D mapping : exploit optimal coverage

Mapping relies on visiting frontiers (known-unknown areas)

→ Repeat assignation robot-frontier : $\min \sum_{(i,j)} \text{cost-Dist}(r_i, f_j)$

[Yamauchi98] (mindist), [Burgard05] (greedy), [Bautin12] (minpos) ..

Sub-problem: **optimal coverage**

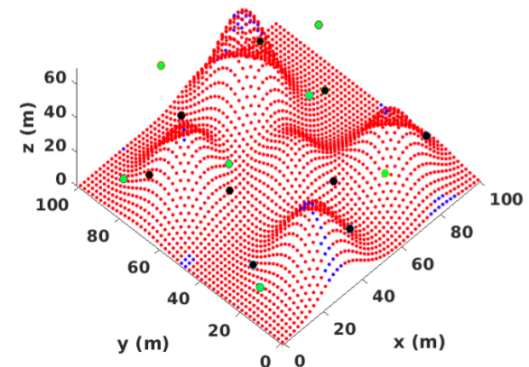
→ deploying robots such as **maximizing the observed area**

CAO (Cognitive-based Adaptive Optimization) [Renzaglia et al. 12]

Local approx. of the **objective** function : $\hat{J}(x_1(t), \dots, x_N(t)) \leftarrow \text{measures (short T)}$

Move : **Stochastic search** on \hat{J}

→ converge to a local optimum



Combining local search and frontier-based app.

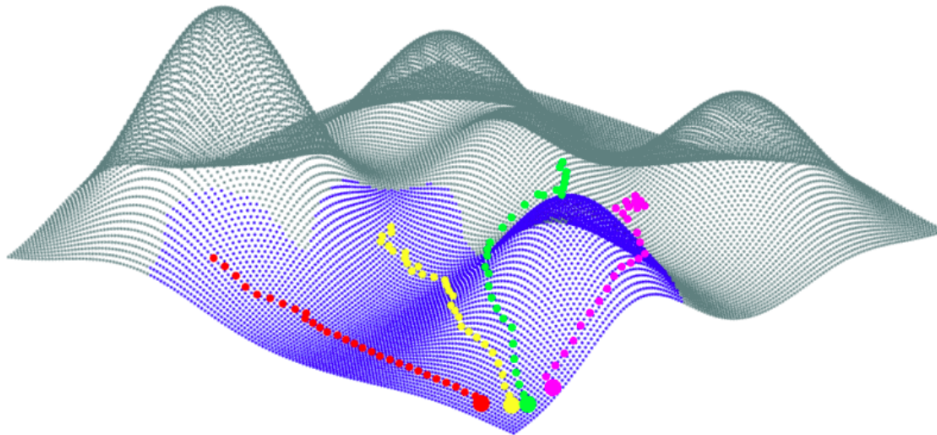
For each robot

Repeat

Follow local search

When a local minima is reached **Move** to the closest frontier

Until no more frontier



Combining local search and frontier-based app.

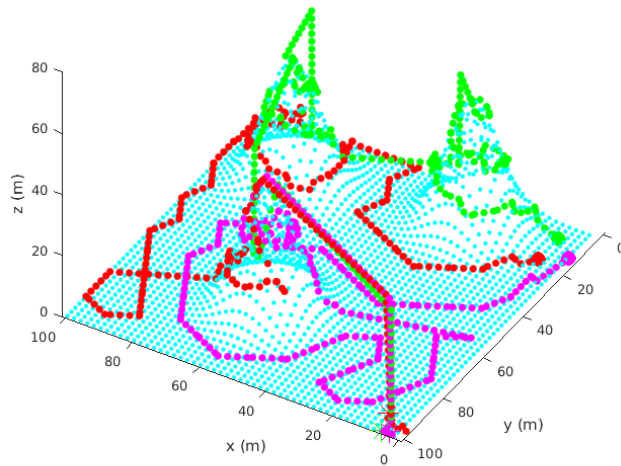
For each robot

Repeat

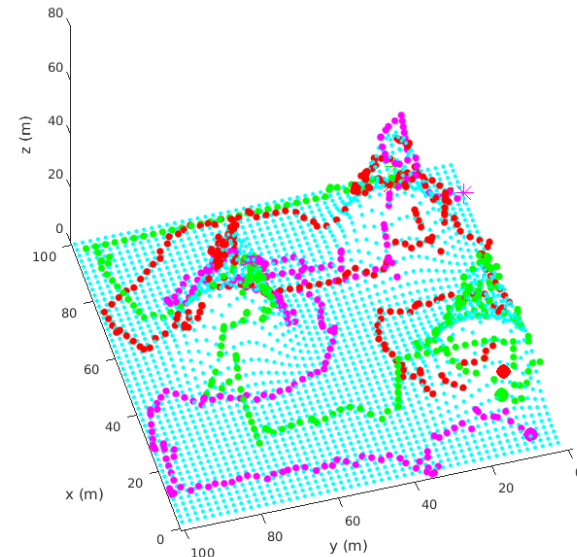
Follow local search

When a local minima is reached **Move** to the closest frontier

Until no more frontier



Only frontiers



Local search + frontiers

Combining local search and frontier-based app.

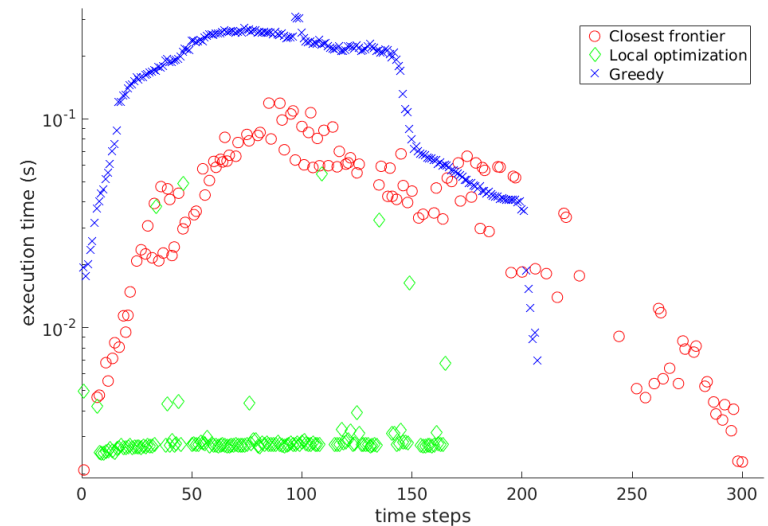
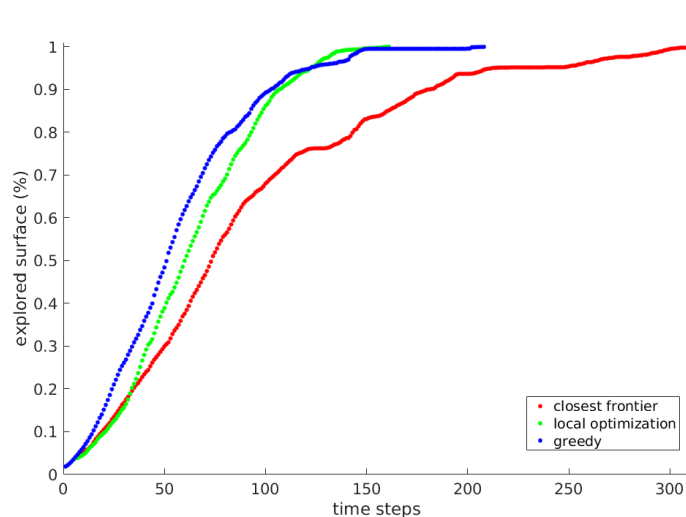
For each robot

Repeat

Follow local search

When a local minima is reached **Move** to the closest frontier

Until no more frontier



Thank you !



<https://team.inria.fr/chroma/>